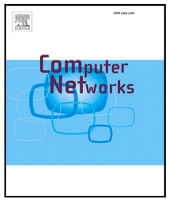
Computer Networks 228 (2023) 109736

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet



Correlation of cyber threat intelligence with sightings for intelligence assessment and augmentation

Po-Ching Lin [a,*], Wen-Hao Hsu [a], Ying-Dar Lin [b], Ren-Hung Hwang [c], Hsiao-Kuang Wu [d], Yuan-Cheng Lai [e], Chung-Kuan Chen [f]

[a] Department of Computer Science & Information Engineering, National Chung Cheng University, Chiayi, Taiwan
[b] Department of Computer Science, National Yang Ming Chiao Tung University, Hsinchu, Taiwan
[c] College of Artificial Intelligence, National Yang Ming Chiao Tung University, Tainan, Taiwan
[d] Department of Computer Science & Information Engineering, National Central University, Taoyuan, Taiwan
[e] Department of Information Management, National Taiwan University of Science and Technology, Taipei, Taiwan
[f] Cycraft Technology, New Taipei, Taiwan

ARTICLE INFO

Keywords:
Cyber threat intelligence (CTI)
Assessment
Augmentation

ABSTRACT

Cyber threat intelligence (CTI) provides the means to rapidly identify and investigate attacks such that the security risks can be addressed. However, few studies have assessed the consistency between the CTI and the observations in the real-world environment (i.e., sightings). Accordingly, this study proposes an approach for assessing such consistency. The assessment process involves finding both false positives (i.e., attacks reported in the CTI, but not observed in the sightings) and false negatives (i.e., attacks observed in the sightings, but not reported in the CTI). The latter are then used to augment the CTI. Several strategies are proposed for assessment and augmentation with a large number of flows in the sightings. For assessment, we first list the characteristic rules for various attacks, and see whether the characteristics of the malicious flows labeled with the attack tags by the CTI match the corresponding rules. We also divide the reported malicious flows into clusters for easier observation. For augmentation, a machine learning framework is employed to identify flows in the sighting with a behavior similar to that of known malicious flows. The attack type and severity of these flows are predicted and used to update the CTI accordingly. The experimental results reveal that among the sightings, over 50% of the flows do not exhibit the behaviors expected from the characteristic rules, but nevertheless appear to be probing or scanning. The proportion of such flows is greater than 90% in the largest cluster for each attack type. When the learning framework is employed, the number of high-severity malicious sources identified in the sighting increases by 156% compared to that reported in the original blacklist. In addition, around 53% of these sources are also considered as potentially malicious by other intelligence sources, and are thus regarded as valid candidates for CTI augmentation.

1. Introduction

Cyber security is regarded by the World Economic Forum as a critical risk that results in significant financial loss [1]. To protect network environments from malicious attack, it is common practice to deploy security systems such as firewalls and intrusion prevention/detection systems to detect or block attacks based on pre-configured rules, attack signatures, and detection models. However, such systems may be unable to identify malicious sources proactively until attacks actually occur. Furthermore, their performance may be limited to inaccuracies inherent in the detection techniques employed. In view of this, cyber threat intelligence (CTI), which contains rich information about threats, such as malicious IP addresses, domain names, hash values, and tactics, techniques and procedures (TTP) of threats, plays an invaluable role in developing more robust defense systems. Moreover, CTI provides a powerful tool for analyzing cyber security events and identifying the threats within them. Thus, network administrators have come to rely increasingly heavily on CTI in recent years as a means of defending against malicious sources by, for example, configuring a firewall with the blacklists provided by CTI to block malicious sources at an early stage of attack (e.g., when a threat actor probes targets for a later attack).

* Corresponding author.
E-mail addresses: pclin@cs.ccu.edu.tw (P. Lin), ejk2013421@gmail.com (W. Hsu), ydlin@cs.nctu.edu.tw (Y. Lin), rhhwang@nycu.edu.tw (R. Hwang), hsiao@csie.ncu.edu.tw (H. Wu), laiyc@cs.ntust.edu.tw (Y. Lai), ck.chen@cycarrier.com (C. Chen).

https://doi.org/10.1016/j.comnet.2023.109736
Received 8 December 2022; Received in revised form 17 March 2023; Accepted 23 March 2023
Available online 29 March 2023

CTI is derived from various sources, including security vendors and expert reports. The intelligence can be in the form of a comment, article, note, tweet, and so on. However, fast-changing threats in the real world, and the limited views of most information sources, may result in inconsistent, inaccurate, and incomplete threat intelligence. Moreover, integrating different sources and formats may lead to information conflict. Thus, as reported by Gao et al. [2], although structured CTI is useful in capturing fragmented views of threats, it still often lacks the capability to uncover complete threat scenarios due to the disconnected nature of the indicators of compromise (IOCs). Furthermore, while unstructured CTI can provide more comprehensive knowledge about threats, it is challenging to devise automated methods for harvesting such knowledge for threat hunting.

In the context of CTI, the term "sighting" refers to observations made in a real-world operational network or system environment, such as network flows, captured packets, and system logs. For example, sighting is performed by logging packets or flow information at a router or switch. The gathered information not only yields valuable insights into the actualities of the CTI in the real world, and but also provides a means of assessing the consistency of the CTI with the actual sightings. Furthermore, by examining the similarity of the activities found in the sightings with those described in the CTI, it is possible to uncover more potential threats through methods such as machine learning. Accordingly, the current study presents a comprehensive approach for assessing and augmenting the CTI based on the sighting observations. In doing so, the study fills an important gap in the extant literature, which generally considers either the assessment of CTI or its augmentation, but seldom both.

In general, the consistency of CTI with the real-world actualities in the networks can be assessed by collecting the NetFlow logs of all the activities reported as malicious by the CTI. However, the sheer volume of these logs precludes exhaustive manual inspection and analysis. Moreover, the logs lack ground truth data for reference purposes. Accordingly, in the approach proposed in the present study, a set of characteristic rules is first constructed for each of the attack types reported by the CTI such that the proportion of reported attacks that really take place in the sighting can be quickly and crudely estimated. Having identified possible malicious behaviors in the sighting, the flows related to each type of attack are clustered and then inspected in accordance with their features. Since the flows in the same cluster are similar one another in some way, it is sufficient simply to sample the flows from each cluster for further investigation. As a result, the manual inspection load is significantly reduced.

As mentioned above, the CTI may be incomplete because of the limited view of the information sources. As a result, the ability of the CTI to detect and predict malicious attacks is inevitably impaired. Consequently, the present study not only proposes the consistency evaluation method described above, but also proposes an approach for detecting potential attacks in the sighting based on their similarity to previous reported attacks and predicting their tags (i.e., attack types) and severity. In particular, the flows associated with the attacks reported by the CTI are first used to train an unsupervised learning model to detect similar flows in the sighting. Having detected such flows, two more learning models, also trained using the malicious flows previously reported by the CTI, are used to predict the tags and severity of the flows. The results are then applied to prioritize the flows for further investigation with the aim of augmenting the CTI if appropriate.

To the best of the authors' knowledge, the present study is the first to correlate the CTI with the full sighting in an operational network for assessing and augmenting the intelligence. Through the assessment and augmentation processes outlined above, this study aims to answer the following questions that were rarely studied in the literature:

- Do the malicious sources in the sighting behave in the ways reported by the CTI?
- Do the tags and severity of the malicious sources marked by the CTI precisely reflect the actualities in the sighting?
- How many network activities in the sighting are similar to those from known malicious sources, and how should their sources be prioritized for augmentation of the CTI?

Addressing the issues can shed light on how the CTI should be interpreted in the sighting of an operational network, and also provide the methods to identify more malicious sources before the CTI is updated. The contributions will be useful for both network analysts and administrators.

The remainder of this paper is organized as follows. Section 2 presents the background to the general CTI and sighting field and reviews previous work on CTI assessment and augmentation, respectively. Section 3 introduces the considered scenario and formulates the related problem statements. Section 4 describes the proposed CTI assessment and augmentation methods. Section 5 introduces the implementation details of the proposed methods. Section 6 presents and discusses the assessment and augmentation results. Finally, Section 7 provides some brief concluding remarks and indicates the intended direction of future research.

2. Background and related work

2.1. Cyber threat intelligence

Farnham [3] divided CTI sources into three categories: internal, community, and external. The internal category incorporates the intelligence gleaned from the inside environment of an organization. By contrast, the community category includes the intelligence shared between multiple organizations that trust each other, such as the Information Sharing and Analysis Centers (ISACs) in multiple European countries, which combine various units and organizations in the governments or private sectors to share intelligence. Finally, the external category contains intelligence which stems neither from the organization itself nor from a community group. Typically, such intelligence originates from open-source intelligence (OSINT) sources [4] and is manifested in many different forms, including reports written by experts, formatted files, essays on cyber security, lists of specific objectives, and so on. However, no matter what form the CTI takes, the information which it provides can help to identify malicious entities such as IP addresses and domain names. The intelligence can significantly improve the detection accuracy of firewalls, intrusion detection, and malware detection.

2.2. Sighting

The term "sighting" refers to the cyber activities observed on the hosts or in the network of the real world. The present study focuses on the particular case of sightings in the network, where the associated information may range from discrete raw packets to higher-level concepts such as network flows, network traffic statistics, and events. Such network traffic is generally collected on the router between the institutional network and external networks because the boundary between them provides a convenient and strategic position for observing external attacks. Network traffic collection is commonly performed at the raw packet level. However, the packet volume in real-world networks is often extremely large, and hence the associated storage cost is extremely expensive. Moreover, inspecting the packets and their payloads may raise important privacy concerns. Consequently, a more scalable and ethical practice, particularly in large networks, is to aggregate the packets into flows for analysis purposes [5]. For example, NetFlow [6] and IP Flow Information eXport (IPFIX) [7] are widely used for network monitoring. The flow data typically contain information such as the source/destination IP addresses/ports, the start/end times, the duration, the number of bytes and packets associated with each flow. Such flow information preserves the major characteristics of the network behavior and is thus invaluable for various purposes

Table 1
Recent CTI-related studies categorized by assessment and augmentation.

Paper	Objective			Input		Output
	Assessment	Augmentation	Aggregation	CTI	Sighting	
No assessment or augmentation (for aggregation only)						
Lee et al. 2018 [10]	×	×	✓	IoC	×	CTI graph database
Gao et al. 2020 [11]	×	×	✓	IoC	×	A CTI modeling and identification system
For assessment only						
Gao et al. 2018 [12]	Trust evaluation	×	×	IoC	×	Threat intelligence graphs
Mavzer et al. 2021 [13]	Completeness, extensiveness, freshness and quality	x	×	IoC	x	66 intelligence items
For augmentation only (some also implement aggregation)						
Huang et al. 2021 [14]	×	TTP	×	MITRE	×	Enrich the knowledge associated with TTP
Gao et al. 2021 [2]	×	IoC	×	OSCTI text	System entities	Threat behavior extraction from OSCTI text
Azevedo et al. 2019 [15]	×	IoC	✓	IoC	×	Quality threat intelligence
Sills et al. 2020 [16]	×	CKG	✓	IoC	×	A repository of known security vulnerabilities
Mavroeidis et al., 2021 [17]	×	Actor	✓	TAL	×	Threat actor type inference and characterization
Suryotrisongko et al., 2022 [18]	×	IoC	✓	IoC	DNS logs	Robust botnet DGA detection
Berady et al. 2021 [19]	×	IoC	✓	IoC	TTP of red team	Advanced persistent graphs
For both assessment and augmentation						
Mills et al. 2021 [9]	Geolocation, IP address and port	blacklist	×	IoC	Dataset from honeypots	An intrusion detection framework
Our work	False positives and false negatives of CTI	IoC, tag, severity	×	IoC	Full sighting in an operational network	Consistency analysis and candidates to be augmented

of traffic analysis, including network measurement (e.g., [8]) and threat detection. Thus, while packet capture can provide more detailed insights into the network traffic, the present study focuses on the problem of collecting and analyzing the network flows in the sighting.

2.3. Related work

Table 1 summarizes the recent related research in the CTI field. With the exception of the present study, only one previous study [9] considered both the assessment and the augmentation of CTI. Moreover, existing studies *do not refer to the sighting or refer only to the network traffic of a limited scope* (e.g., that from a red team or honeypots). In other words, the present study provides the first reported attempt to jointly assess and augment the CTI based on the full sighting in an operational network. The details and scope of the other studies reported in Table 1 are presented in the following.

Gao et al. [12] collected CTI from several threat intelligence sharing platforms, and used the information to construct a threat intelligence graph. The graph was then as the basis for an automatic trustworthiness calculation mechanism. Mavzer et al. [13] performed four tabletop exercises to evaluate various CTI sharing tools such as ECHO – Early Warning System (E-EWS) in terms of 66 intelligence items. As in the present study, Mills et al. [9] utilized CTI to label the flow data in the network. However, the study focused mainly on the problem of identifying the geolocation and IP addresses of actual attacks rather than the feasibility of utilizing the CTI to predict potential future attacks. Moreover, while the present study employs an unsupervised learning approach to detect previously unreported malicious IP addresses and predict the attack type and severity of the associated flows as a means of augmenting the CTI, the study in [9] employed a supervised learning approach to carry out intrusion detection.

Several studies have considered the problem of aggregating CTI from multiple sources to produce a single piece of CTI. For example, Lee et al. [10] presented a standardized management structure for integrating the intelligence received from multiple sources into a graph database and then sharing the CTI outside. Gao et al. [11] developed a practical system called HinCTI for aggregating and modeling CTI from different sources and generating high-quality CTI such as the threat type. Azevedo et al. [15] proposed a platform referred to as PURE to create quality threat intelligence through OSINT sources via clustering and correlation mechanisms. Sills et al. [16] used CTI to construct a Cybersecurity Knowledge Graph (CKG) which was then augmented to produce a higher quality graph through the integration of multiple resources. Mavroeidis et al. [17] used the Intel Threat Agent Library (TAL) to generate highly contextual, explicable, processable, and shareable threat actor intelligence. Huang et al. [14] combined OSINT and various resources on the MITRE website to develop a system, designated as MAMBA, to recognize threat techniques and discover TTP and malicious behavior. The proposed method identified malicious behavior through the use of deep learning. However, the quality of the detection results was heavily dependent on the reliability of the aggregated CTI sources, and *aggregating multiple CTI sources is beyond the scope of our work*.

Several studies have used sighting data such as DNS logs, system entity records, and the TTP of the red team to facilitate threat detection and CTI investigation. Gao et al. [2] proposed a system referred to as ThreatRaptor for facilitating threat hunting in computer systems using open-source CTI (OSCTI) text. A model was additionally proposed for detecting the potentially malicious traffic produced by domain generation algorithms (DGAs). Suryotrisongko et al. [18] showed that CTI methods face difficulties in detecting DGA-based botnets using blacklists. Thus, a method was proposed for detecting such traffic using

Table 2
Notations.

Category	Notation	Description
NetFlow	$F = \{f_1, f_2, \ldots, f_m\}$	NetFlow logs, in which f_i stands for a flow.
	V_e	Set of external hosts
	V_i	Set of internal hosts
CTI	$B = \{b_1, b_2, \ldots, b_n\}$	Set of blacklisted IP addresses from the CTI
	$B^+ = \{b_1^+, b_2^+, \ldots, b_l^+\}$	Set of malicious IP addresses found in the augmentation
	$T = \{t_1, t_2, \ldots, t_k\}$	Set of attack tags in the CTI
	$S = \{1, 2, \ldots, s\}$	Range of severity in the CTI
Model	D_M	Malicious flow detector
	P_T	Tag predictor
	P_S	Severity predictor

the statistical features of a large number of DGA families. In addition, the interpretability of the detection results was enhanced by combining explainable AI and OSINT. Berady et al. [19] presented a model for analyzing cyber-attacks from the perspectives of both the attacking red team and the defending white team, respectively, in order to better understand the TTP of both sides and identify the origins of objects which then become IoCs.

In summary, although the existent studies may utilize dedicated security appliances like us, as well as some graph or learning models in their designs, the appliances and models are used to correlate the CTI from multiple sources, or to label the specific traffic from the CTI. Unlike the present study, the existent studies do not assess the consistency between the CTI and the actualities in the sighting of an operational network, and augment the CTI accordingly.

3. Scenario description and problem statements

In this study, it is assumed that a security device is deployed at the typical location of a firewall, and is able to observe all the network traffic between the internal and external networks. It is further assumed that the security device has access to the CTI and can refer to a blacklist of IP addresses $B = \{b_1, b_2, \ldots, b_n\}$, and choose to either block or log the network traffic accordingly. For convenience, the device is referred to simply as a CTI device hereafter. The CTI device is assumed to have the ability to mark each blacklisted IP address, b_i, with one or more tags from a set of pre-defined attack tags T to indicate the type of attack, such as botnet, and its severity in the range S (e.g., between 1 to 10). In addition, to obtain the raw network traffic from the sighting, it is assumed that a NetFlow collector (located on a border router, for example) observes the same network traffic as that seen by the CTI device, and sample every packet in the network traffic to generate a set of NetFlow logs $F = \{f_1, f_2, \ldots, f_m\}$. The notations used throughout this study are listed in Table 2.

3.1. Problem statement for CTI assessment

The first objective of the present study is to design a method to assess the consistency of the CTI with the real-world sightings. Given the NetFlow logs F and blacklist of malicious IP addresses, B, the assessment process sets out to determine whether the flows in F whose source hosts are included in B really exhibit the characteristics of the attack types indicated by their tags and severity labels in the CTI. The more the characteristics observed in the sighting conform to the tags and severity of the blacklisted IP addresses reported in the CTI, the more consistent the CTI is with the sighting. In general, inconsistencies between the CTI and the sighting may arise for two main reasons. First, the flow is from a normal host but the source host is listed in B, or the flow is erroneously labeled with an attack tag and associated severity in the CTI (i.e., the flow represents a false positive). Second, the flow is from a malicious host but the host is not listed in B, or the flow (which is malicious) is not labeled with an attack tag or associated severity in the CTI (i.e., the flow is a false negative). In practice, the assessment process is extremely challenging since there are no ground truth labels in the real network to indicate whether or not the flows behave consistently with their tags/severity labels in the CTI. Furthermore, given thousands of flows or even more per minute in a typical network, manually determining whether the nature (i.e., benign or malicious) of each flow is a next to impossible task.

3.2. Problem statement for CTI augmentation

The second objective of the present study is to find potentially malicious network flows in the sighting that are not associated with any of the blacklisted IP addresses reported in the CTI (i.e., false negatives on the CTI device), and to augment the CTI accordingly. Formally, given the NetFlow logs, F, and blacklisted IP addresses, B, together with their attack tags and severity labels, the augmentation process aims to determine the set of malicious source IP addresses in F but not in B (referred to henceforth as B^+), and to update the tags and severity associated with each member in $B \cup B^+$ as required. Note that some of the blacklisted IP addresses may not be tagged with any attack type in the CTI; thus, the augmentation process also includes marking such IP addresses and updating their severity. As for the CTI assessment process, the augmentation process is challenged by the absence of ground truth labels for the network flows in the sighting. Consequently, the present study utilizes the information associated with the reported malicious flows and the tags of the flow sources in the CTI to construct three prediction models to facilitate the augmentation process, namely (1) a malicious flow detector D_M, (2) a tag predictor P_T, and (3) a severity predictor P_S. Collectively, the three models provide the means to find previously unseen malicious flows in the sighting, predict their tags, and update the severity of the respective IP addresses in the set $B \cup B^+$. Besides the intelligence sharing platform used in the present study (see Section 6.1), the augmented intelligence items (e.g., blacklisted IP addresses) can be formatted to be incorporated into other platforms such as the malware information-sharing platform (MISP),[1] e.g., as a reported event in the MISP.

3.3. Scope and limitations

CTI involves a wealth of information, including malware hashes and the IP addresses and domain names of malicious hosts. However, collecting and analyzing all of the available information in the sightings in order to perform the assessment and augmentation processes described above require the use of enormous resources and complex processing. For example, it may be necessary to deploy honeypots and system monitoring tools on many user hosts in order to collect sufficient information to detect and collect malware. Furthermore, it may be necessary to collect not only the packet headers, but also the packet payloads, which is time-consuming and resource intensive and is complicated by the many encryption protocols in use nowadays. Finally, performing a deep analysis of such information in the sighting may violate user privacy laws. Thus, while involving more intelligence in the sighting to perform the assessment and augmentation tasks is an appealing option to obtain more insight, the present study focuses on the more practical case of analyzing L3/L4 information for scalable processing and in order to avoid such privacy concerns and better match the capabilities of the devices used in our network (see Section 6.1).

[1] https://www.misp-project.org

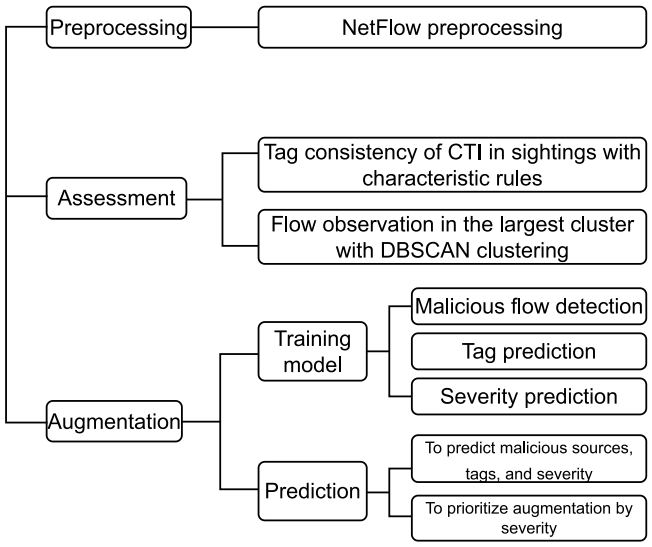


Fig. 1. Basic workflow of proposed CTI assessment and augmentation framework.

4. Assessment and augmentation methods

4.1. Design overview

Fig. 1 presents an overview of the proposed CTI assessment and augmentation framework. In the NetFlow data used in the preprocessing stage, each flow contains various fields, such as the source and destination IP addresses/ports, the numbers of bytes and the packets, and the timestamp. In the present study, the interactions between the hosts in the internal and external networks in the NetFlow sighting data collected over a certain period of time are modeled as a bipartite graph. The statistical features of the nodes and edges in the graph are determined based on the NetFlow data. Meanwhile, the blacklisted IP addresses, together with their tags (if any) and severity, are obtained from the CTI logs maintained by the CTI device. The statistical sighting information and CTI data are then provided as inputs to the assessment and augmentation processes, as described in the following.

After the network activities in the sighting have been modeled, the consistency of the CTI with the sighting is evaluated. Note that without any ground truth data, the consistency can only be estimated (rather than precisely quantified), given the huge effort of manually examining a large number of flows in a real network. To ease the CTI assessment task and identify false positives, *characteristic rules* are thus derived for each tag (i.e., each type of attack) in the CTI to describe the characteristic behavior of most network flows associated with sources labeled with this tag (see Table 5). The rules then provide a convenient approach for quickly identifying those flows whose sources are supposed to be unlikely marked with the corresponding tag. The network flows whose sources are marked with the same tag are then divided into clusters by *DBSCAN clustering*. Through the use of the clustering process, the behaviors of the corresponding flows associated with each tag can be investigated using a simple sampling process rather than by inspecting each flow individually. Consequently, the investigation load is significantly eased, particularly in the case of clusters containing a large number of flows.

One of the main objectives of the CTI assessment process is the identification of false negatives, i.e., malicious sources in the sighting which are regarded as benign in the current CTI. In the present study, these false negatives are considered as potential candidates to augment the CTI so as to improve its robustness in the future. In the proposed augmentation approach, the current CTI is first used to label the known malicious flows in the sighting, and these labels are used to train a one-class support vector machine (one-class SVM or OCSVM) model, referred to as the *malicious flow detector*, to identify additional malicious sources in the sighting. Since such malicious sources, and some blacklisted IP addresses, do not have attack tags, or their severity values are presently underestimated in the CTI, two further models are also trained, namely a *tag predictor* and a *severity predictor*, to update the source/IP tags and severity, respectively. The resulting list of malicious IP addresses, tags, and severity values are then taken as potential candidates for CTI augmentation. Note that not all of the findings are added to the CTI because of the inherent inaccuracy of the original CTI and the first learning model. Consequently, the predicted severity values are used as a reference in prioritizing which particular findings should be added to the CTI. The validity of the prioritization results is then further validated by reference to public intelligence and intrusion prevention system (IPS) logs for the sighting (if available).

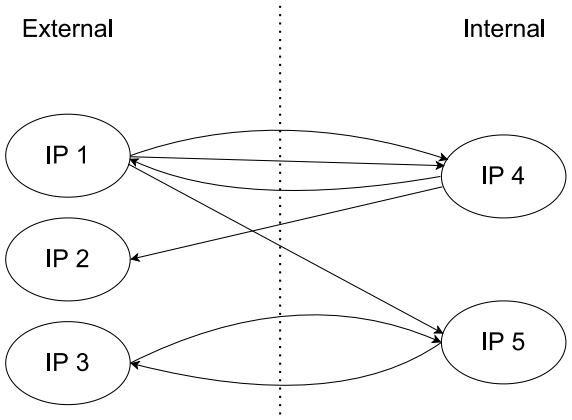


Fig. 2. Illustrative example of directed bipartite multi-graph.

4.2. Preprocessing

Since viewing the individual flows separately is insufficient to characterize the network behavior of the involved hosts, a directed bipartite multi-graph is built from the NetFlow records collected each hour to model the interactions between the internal and external hosts. In the graph, the nodes represent the hosts whose IP addresses appear in the NetFlow logs and are identified as either internal nodes (belonging to the internal network) or external nodes (belonging to an external network). Meanwhile, the edges represent the flows between the various hosts. For modeling purposes, each edge is characterized by four fields of the flow record, namely (1) the date first/last seen, (2) the source/destination address/port, (3) the byte count, and (4) the packet count.

Fig. 2 illustrates the directed bipartite multi-graph using a trivial example involving IP 1, IP 2 and IP 3 as external hosts, and IP 4 and IP 5 as internal hosts. The edges represent the interactions between the two sets of hosts over a certain period (e.g., 1 h). It is seen that some interactions are one-sided, e.g., from IP1 to IP 5, while others are two-way, e.g., between IP 3 and IP 5. Furthermore, for some interactions (e.g., IP1 to IP5), only one flow occurs during the observation period, whereas in other interactions (e.g., IP 1 to IP 4), multiple flows take place.

Having constructed the bipartite multi-graph, the time-coupled process of the flows over the observation period is modeled by extracting 19 features from the graph in accordance with the guidelines of Krishnamurthy et al. [20]. The features relate to the numbers of packets, bytes, and port numbers associated with or between the nodes, and may be related to either individual nodes or across flows (see Table 3 and Table 4, respectively).

Table 3
Features of nodes extracted from the bipartite multi-graph.

Features	Description
packets_sent bytes_sent	The total number of packets/bytes sent from a host (i.e., the source IP addresses in the flows.)
packets_recv bytes_recv	The total number of packets/bytes received by a host (i.e., the destination IP address in the flows.)
packets_sent_to_port bytes_sent_to_port	The total number of packets/bytes sent from a port (i.e., the source ports in the flows.)
packets_recv_to_port bytes_recv_to_port	The total number of packets/bytes received by a port (i.e., the destination ports in the flows.)
distinct_to	The total number of nodes that have an edge to a node.
distinct_ports	The total number of ports that have an edge to a node.
avg_packets_sent_size	The average bytes in a packet sent from a host.
avg_packets_recv_size	The average bytes in a packet received by a host.

Table 4
Features across flows extracted from the bipartite multi-graph.

Features	Description
packets_sent_to_target bytes_sent_to_target	The total number of packets/bytes sent to a target host from a source host in the flows.
packet_recv_from_target bytes_recv_from_target	The total number of packets/bytes received from a source host by a target host in the flows.
n_entries_to_port	The number of flows between two ports.
n_entries_to_target	The number of flows between two nodes.
distinct_ports_to_target	The number of destination ports between two nodes.

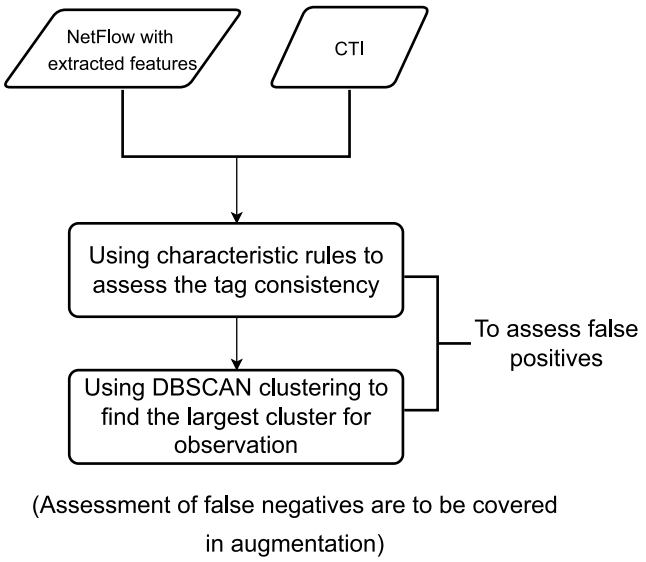


Fig. 3. Flowchart of CTI assessment process.

Table 5
Characteristic rules of attack types.

Types	Rules
botnets	The source or destination port is related to remote services.
exploits	A client generates at least three flows with the mean packet size larger than 64 bytes.
Tor	The port of the external host is a common Tor port.
phishing	The source or destination port is related to mail or web.
ransomware	Manual review because of sparse flows.
malware	The source or destination port is a common port related to malware.
spam	The source or destination port is related to sending/receiving mail.
cryptomining	The external port is a common mining port.
scanner	To more than 10 IP addresses with the same port. To the same IP address with more than 10 ports.

4.3. CTI assessment based on netflow

The purpose of the CTI assessment process is to determine the false positives and false negatives in the sighting as compared with the CTI report. As described in Section 4.1, the false negatives are regarded as potential candidates for augmenting the CTI. Hence, the details of the false negative identification process are presented later in Section 4.4, which describes the entire CTI augmentation process. Accordingly, the remainder of this subsection focuses on the problem of finding the false positives in the sighting. The CTI assessment process for false positives involves two mechanisms, namely *characteristics rules* and *DBSCAN clustering*, where both mechanisms refer to the features extracted from the directed bipartite multi-graph described in Section 4.2. Fig. 3 illustrates the basic procedure of the CTI assessment process.

4.3.1. Characteristic rules

To facilitate the CTI assessment process, a set of characteristic rules are derived relating to the connected ports and activities of the IP addresses for each attack tag labeled by the CTI. The rules then provide a convenient approach for rapidly identifying potential false positives among the reported malicious IP addresses in the CTI device (i.e., the flows from these addresses do not match the corresponding rules). Table 5 summarizes the attack tags considered in the present study and their characteristic rules. It is seen that the rules are simple, and can thus be easily implemented in the assessment process. For example, some of the rules consist simply of common ports associated with particular types of attack, or the applications in which the associated attacks most commonly occur [21]. In general, it is difficult to enumerate a set of rules which exhaustively and precisely covers all possible characteristics of a particular attack type since each attack type may consist of multiple attack techniques (e.g., there are various botnet variants in the wild). In view of this, and since the aim here is to focus only on a crude evaluation of possible false positives in the assessment, the rules in Table 5 intended only to describe the general characteristics that the attacks of a certain tag typically display. The details of the characteristic rules for each tag are described in the following.

- botnet: The assessment process considers two botnet scenarios: (1) the host is a command and control (C&C) server, and (2) the host has been compromised and is a bot. Possible ports likely associated with C&C channels include those commonly used for remote services, IRC channels, HTTP, and printers [21]. The ports associated with known botnets are also considered. Thus, the ports considered in the characteristic rule for botnets are specified as follows:
  - related to remote services, including those found in the sighting: 21, 22, 23, 135, 177, 530, 903, 2049,3389, 2181, 5500, 5984, 6000, 6008, 7687, 9200, 9600, 11,211
  - IRC channels: 6667, 6697
  - HTTP, HTTPS: 80,443
  - printer ports: 35, 515, 631, 2081, 2991, 3396
  - found by searching ‘botnet’ in the speed guide: 636, 989, 990, 992, 994, 995, 3269, 5645, 7779, 8080, 8443, 13,620, 16,464, 16,465, 16,470, 16,471, 21,315, 21,810, 22,292

- exploit: Exploiting attacks generally requires several attempts to succeed. Hence, if a host initiates at least three flows and the mean packet size is larger than 64 bytes (i.e., carrying a non-empty payload), it is regarded as a potential exploitive host.
- Tor: The assessment process considers the case where an internal host connects to an external Tor network. The ports most commonly used for Tor include 81, 82, 9001, 9030, 9040, 9050, 9051, 9150.
- Phishing: Several common phishing scenarios are considered, including external hosts sending phishing mail to internal mail servers or a web service (i.e., web phishing), or internal hosts accessing a phishing web service. Thus, the characteristic rule is defined in terms of the ports most commonly associated with sending and receiving mail, or accessing web services, i.e.,
  - Sending/receiving mail: 25, 57, 587, 465, 2525, 109, 110.
  - Related to web: 80, 443, 8080.
  - Obtained from searching for 'phishing' in the speed guide: 85, 880, 4903, 6180, 25,080.
- Ransomware: Such attacks are identified through manual inspection only due to the sparseness of such flows in typical sightings.
- Malware: A total of 119 ports commonly associated with malware, Trojan, worms, and other malware-related attacks are considered [21].
- Spam: As for phishing attacks, several common spamming scenarios are considered, such as external sources sending spam mail to an internal mail server or web service. The characteristic rule is thus configured in terms of the ports most commonly associated with sending and receiving mail, or accessing web services, i.e.,
  - Sending/receiving mail: 25, 57, 587, 465, 2525, 109, 110.
  - Related to mail: 25, 587, 465, 2525.
  - Obtained from searching for 'spam' in the speed guide: 25, 135, 559, 1025, 1026, 1080, 2568, 2599, 2703, 3355, 5190, 6019, 6277, 8910, 9040, 1133, 2113, 3311, 3342, 5080, 65,506.
- Cryptomining: The assessment process assumes that when the external port is a common port for mining pools, the flow is likely to be a cryptomining flow. Thus, the characteristic rule consists of ports 443, 4444, 6641, 6642, and 8333, together with ports derived from the web speed guide, including 8123, 8124, and 8125.
- Scanner: Two scanner scenarios are considered: IP scanning and port scanning. The corresponding characteristic rule is thus identified as follows:
  - IP scanning: A source host tries to sending packets to more than 10 internal hosts with the same port.
  - Port scanning: A source host tries to send packets to more than 10 ports with the same host.

The characteristic rules described above exploit the fact that different attack types usually exhibit different characteristics. In the CTI assessment process, the flows obtained over different time frames (i.e., one hour, one day, and 12 days in our observation (see Section 6.2.1)) are observed and evaluated with reference to the characteristic rules. If an observed IP address is tagged with one of the attack types listed in Table 5, a check is made as to whether at least one flow from this IP address matches the corresponding rule(s) of the attack type within the considered time period. If at least one match is obtained, the assessment process assumes that the IP address "may" exhibit the behavior of the associated attack type; otherwise, the IP address is very unlikely associated with the attack type (i.e., it is likely a false positive).

However, some flows which do not match the rules may still be an attack (e.g., an attack with an unusual port number). Consequently, the true number of false positives may be less than that predicted through the rule matching. Nonetheless, the rules in Table 5 are deliberately defined as loosely as possible to reduce such cases, and hence it is reasonable to assume that the overestimation of false positives is small in practice. However, the rule-matching approach may also underestimate the number of false positives since no formal check is actually made in the assessment process as to whether the reported malicious flows that match the rules are truly positive instances. Because of the looseness of the rules, it is likely that the underestimation of the false positives is more significant than their overestimation. As a result, it is inferred that the overall estimation is roughly the lower bound of the true number of false positives.

4.3.2. DBSCAN clustering

To verify the results of the rule-matching process and better understand whether the tags from the CTI are truly consistent with the observations in the sighting, the network flows in the present study are also analyzed by manual inspection. As described earlier, it is impractical to manually inspect every flow because of the overwhelming number of flows in the sighting. Thus, in the CTI assessment process, a clustering approach is employed to group the similar flows per tag such that representative flows can then be sampled at random for manual inspection. Among the popular clustering algorithms available, k-means and DBSCAN are two of the most commonly used. The present study deliberately adopts the latter method since it is density-based and can find clusters of arbitrary structures without being affected by noise (i.e., noise is treated simply as an outlier). Consequently, DBSCAN is an effective means of obtaining the clusters that require manual inspection.

As described in Section 4.2, the flow features are extracted from the bipartite multi-graph. DBSCAN is then run for each attack tag such that all of the flows associated with each tag (i.e., attack type) are clustered in accordance with their features. Following the clustering process, the flows in each cluster are randomly sampled, and their features and activities can be further examined manually. Since the number of clusters is much less than the number of flows, such an approach provides a convenient and efficient means of understanding the activities involved in the major clusters of each attack tag.

4.4. CTI augmentation based on NetFlow

The CTI augmentation process involves finding malicious flows in the sighting that have not been identified by the CTI device (i.e., false negatives), labeling these newly-found malicious flows (as well as the known malicious flows without tags) with tags, and also predicting their severity values. To facilitate the augmentation process, three learning models are trained to predict the malicious flows, their tags and their severity. As described in Section 4.2, a bipartite multi-graph is created every hour, and the flow features are extracted from it for assessment and augmentation purposes. Fig. 4 illustrates the main steps in the augmentation process. In the training stage, the features of the flows with external blacklisted IP addresses are used to train the three models, namely a single *malicious flow detector* based on one-class SVM, multiple *tag predictors* (one for each attack type) also based on one-class SVM, and a single *severity predictor* based on a random forest regressor. In the subsequent prediction stage, the trained models are used to predict malicious flows and identify their IP addresses, predict the tags of the associated IP addresses, and predict their severity, respectively.

An assumption is made that the hosts that generate malicious flows (i.e., the hosts blacklisted in the CTI device) are likely to be compromised but may still generate many normal flows as a result of their normal usages. It is noted that the flows originating from the external and internal blacklisted sources are observed quite differently in the sighting. In particular, all the flows with Internet access from the internal blacklisted sources can be collected at the border router, regardless of their nature (normal or malicious). However, for the

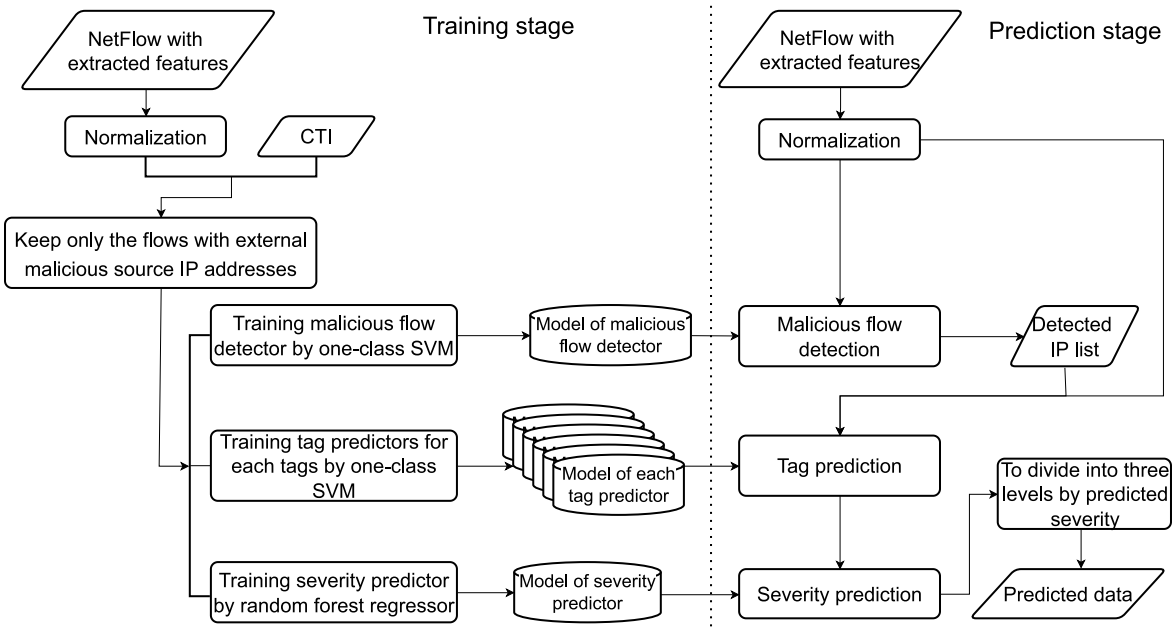


Fig. 4. Flowchart of CTI augmentation process.

external blacklisted sources, almost only the malicious flows produced by these sources are observed in the sighting since their normal flows typically do not appear in the sighting (unless they happen to access the services in the internal network as part of normal usage, which is unlikely). Accordingly, only the flows from external blacklisted sources are used to train the learning models since these flows can be reliably considered to be malicious.

The three learning models operate at the level of network flows rather than IP addresses. Accordingly, to obtain the predicted malicious IP list shown in Fig. 4, an assumption is made that if more than half of the flows from a host are predicted as malicious (or normal), the corresponding IP address can also be predicted as malicious (or normal).

4.4.1. Malicious flow detection model

The malicious flow model is trained with the characteristics of known malicious flows, and then tested with flows in the sighting to see whether any of these flows behave similarly to known malicious flows. Since the CTI device reports a list of malicious flows associated with known blacklisted IP addresses from the CTI, a one-class SVM model is trained using these flows to predict the flows in the sighting. It is noted that the training set contains only known malicious flows from the CTI device, and it is nearly impossible to ascertain whether all the flows not reported by the CTI are false negatives or are indeed truly normal. Hence, one-class SVM, which is trained with only one class of instances, is ideal for the present purposes. In practice, other anomaly detection models are also available for similar classification problems, such as isolation forest. However, a preliminary investigation revealed that these methods had a poorer accuracy or lower efficiency than the one-class SVM method, and hence they were discarded. As shown in Fig. 4, prior to training, the NetFlow data collected in each hour were normalized by min–max normalization. The flows with blacklisted source IP addresses were then used to train the model, as described above.

4.4.2. Tag prediction models

The tag prediction models are also trained using one-class SVM. Since a blacklisted IP address may be marked with multiple tags, a separate model is trained for each tag (i.e., each attack type). Following the malicious flow detection process, only the malicious source IP addresses remain. Thus, only the flows from the blacklisted source IP addresses are normalized for the tag prediction models. After normalization, the flows with blacklisted source IP addresses are divided into different sets according to their tags. If a flow is marked with multiple tags, it is duplicated in multiple sets with these tags. The prediction models for different tags are trained with the flows in the respective sets. The trained models are then used to predict whether each flow in the sighting should be associated with a particular tag or not.

4.4.3. Severity prediction model

The severity value is numerical (i.e., S=1~10) rather than categorical. Hence, the severity prediction model is trained using the random forest regressor, which has good performance empirically. Moreover, in contrast to the other models, the NetFlow data are not normalized prior to training since the regression method is unaffected by the range of the data values. The model is trained using the 19 features extracted from the graph as data and the severity values from the CTI as labels. The trained model is then used to predict the severity of each flow in the sighting.

Having trained the three models, the prediction stage is conducted as follows:

1. The malicious flow model detects the malicious flows in the sighting. If more than half of the flows from an IP address are detected as malicious, the IP address is also inferred to be malicious.
2. Taking the list of malicious IP addresses as an input, all of the flows coming from these IP addresses are processed by the tag prediction model. If more than half of the flows coming from an IP address are predicted to be associated with a particular tag, the IP address is labeled with that tag.
3. The flows coming from IP addresses that have at least one predicted tag are processed by the severity model to predict the corresponding severity values. The average severity value of all the flows from the same IP address is then evaluated as the severity label for the IP address.

5. Implementation

5.1. Cyber threat intelligence solutions

In the present study, the CTI source was provided by a proprietary intelligence sharing platform, which integrated global CTI collected from around 20 CTI sources. One of the main sources was globally reputable cyber security organizations, which were accessed to obtain reputation data. Another source was OSINT, which were used to acquire a wealth of CTI information, such as security reports. Overall, the platform provided a rich CTI database, in which the threat indicator feeds included whois, DNS, certifications, and history records. The CTI on the platform therefore represents the results from multiple common CTI sources, rather than those simply of a single product from a technical company. Moreover, the platform can be used to search for IP addresses, domains, URLs and so on. The possible search results also include information such as basic, whois, and passive DNS. The APIs for various searching functions were incorporated directly within the platform. For example, the most commonly used function in the present study was that of looking up the basic data associated with IP addresses, such as their reputation, various scores, and nature (benign or malicious) according to the CTI.

A proprietary intelligence-based firewall was used as the firewall system based on the CTI provided by the intelligence sharing platform. In particular, the former cooperated with the latter to update its blacklist every hour and to implement active defense measures accordingly. Both the CTI data from the intelligence sharing platform and the blacklist from the intelligence-based firewall were used as references for the assessment and augmentation processes. The firewall system identified malicious flows through an inspection of their IP addresses, domains, and so on. The flows were mirrored from the router to the firewall system, which generated the logs of the malicious flows according to its blacklist on the mirrored flows. Moreover, the firewall system could be configured to block inline malicious flows for defense if deployed in the packet path (despite not the configuration in the present study).

The firewall system generated the logs that contained the detection time, the source IP/port, the destination IP/port, the protocol, and so on. The blacklist was obtained from the logs, and a report was then manually generated from its dashboard. The report included detailed statistics on malicious activities during the observation period, such as the types of attacks detected and the external networks from which they originated.

5.2. Tools and libraries

The NetFlow records were collected and processed using nfdump [22], which incorporates the following main tools:

- `nfcapd` – NetFlow collector daemon
- `nfdump` – to process the collected NetFlow records
- `nfanon` – to anonymize the NetFlow records
- `nfexpire` – to expire the old NetFlow data
- `nfreplay` – to perform Netflow replay

In the present study, only two of the tools, `nfcapd` and `nfdump`, were used. The former tool was used to capture the NetFlow records from the router, while the latter was used to process the captured records.

Scikit-learn [23] is a well-known machine learning module built on the Python modules NumPy, SciPy, and matplotlib. Scikit-learn is open-source and commercially usable, and provides a rich set of learning functions, such as classification, regression, clustering, and dimensionality reduction. The present study employed the following scikit-learn functions:

- One-class SVM – used to implement the malicious flow detection module to determine the nature of the flows (benign or malicious) and the tag prediction module to ascertain the tag (attack type) of each flow in accordance with its flow characteristics (see Section 4.4).
- Random forest regressor – used to implement the severity prediction model to determine the severity of each attack flow (see Section 4.4).
- DBSCAN – used to cluster the flows with the same tag in order to facilitate manual inspection (see Section 4.3.2).
- Some preprocessing functions used to perform data normalization (see Section 4.4).

Finally, `rsyslog`, an open-source utility on Unix for filtering logs, was used to capture the logs forwarded by the intelligence-based firewall.

6. Evaluation

6.1. Experimental setup and parameters

The evaluation experiments were performed using the internal network of the Department of Computer Science and Information Engineering, National Chung Cheng University, Taiwan. The network comprised five /24 subnets. The volume of network traffic between the internal and external networks was around 18 million flows per day during the experimental period. The flows contained approximately 130 to 160 million packets with a total volume of approximately 1TB. The following devices were deployed to obtain the flow records and facilitate the CTI assessment and augmentation processes.

1. Router: A high-end router (Cisco Catalyst 9500) connected the internal network of the department to the outside world. The NetFlow function was enabled on the router to obtain the NetFlow records F. Every packet through the router was sampled for fidelity.
2. Intelligence-based firewall: The flows from the router were mirrored to the firewall, which generated blocking logs corresponding to the mirrored flows. Based on these logs, a blacklist B was created to record the malicious IP addresses blocked by the firewall over the corresponding period. Moreover, the dashboard was used to generate a report based on the information received from both the firewall and the proprietary intelligence sharing platform.
3. Intelligence sharing platform: The intelligence sharing platform integrated the various community and external CTI data into a single database. The intelligence-based firewall obtained the CTI from the platform, and automatically updated its blacklist according to the intelligence received. The APIs built into the platform were used to obtain the data required for CTI assessment and augmentation purposes.
4. Record server: The server received the logs from the intelligence-based firewall using `rsyslog` and the NetFlow data from the router using `nfdump`.

Fig. 5 illustrates the relationship between the various devices. A solid line indicates a physical connection, while a dotted line represents data transfer. It is noted that the assessment and augmentation processes were carried out offline, and thus, the processes themselves did not bring any latency to the operational network. Nonetheless, the proposed framework caused some additional load to the router due to the NetFlow collection and traffic mirroring as presented in Fig. 5. However, the load to the router was lightweight in the configuration, with the CPU utilization of the router merely around 10∼20%. Accordingly, the operational network still worked quite smoothly, without perceivable latency reported by the ordinary users in the operational environment during the experiments.

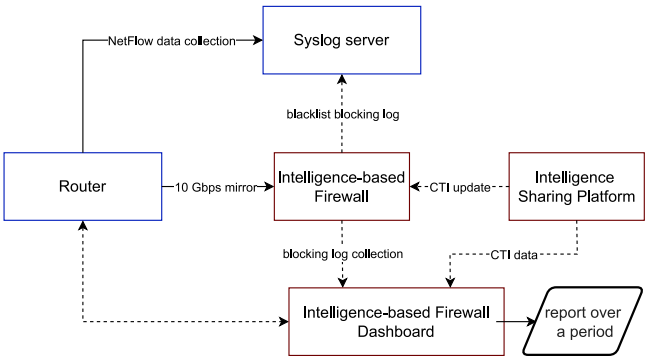


Fig. 5. Relationship between devices in experimental evaluation system.

6.1.1. NetFlow data

The flow records involved around 400 internal IP addresses and 30~200 thousand external IP addresses every hour. Around 100 to 200 malicious IP addresses were reported by the intelligence-based firewall per hour, and around 100 to 600 malicious IP addresses were reported by the firewall each day. The data used in the various experiments are described as follows.

- Characteristic rules: The 12-day data collected from 2021/10/04 to 2021/10/15 were used, where the data involved nearly 2 million flows.
- Clustering: The one-day data collected on 2021/10/04 were used. The data volume is presented in Fig. 7.
- Model training: The 15-day data collected from 2021/10/1 to 2021/10/15 were used, where the data involved around 2 million malicious flows. There were around 0.5 to 1.5 million flows for most of the tags except tor, cryptomining and ransomware tags, which had around 20~40 thousand flows (tor and cryptomining) and a single flow (ransomware), respectively.
- Model testing: The testing data comprised 8 days of malicious flow data (around 1.3 million flows), 1 day of normal DNS data (approximately 1.7 million flows), and 15 days of Google index flow data (around 1.3 million flows).
- •Model evaluation: The three models were combined and evaluated using six hours of data collected from 0 AM to 6 AM on 2022/02/28.

6.1.2. Metrics

Assume that for the problem considered in the present study, a positive instance means a malicious instance. Thus, true positive (TP) means that the prediction of an instance is positive, and the instance truly is positive. Similarly, false negative (FN) means that the prediction of an instance is negative, but the instance is in fact positive. False positive (FP) means that the prediction of an instance is positive, but the instance is actually negative. True negative (TN) means that the prediction of an instance is negative, and the instance truly is negative.

The performance of the one-class SVM models was evaluated using two metrics, namely (1) the false-negative rate (FNR), defined as $\frac{FN}{TP+FN}$, to evaluate the ability of the model to detect malicious instances correctly, and (2) the false-positive rate (FPR), defined as $\frac{FP}{TN+FP}$, designed to evaluate whether the detected instances are truly malicious. For the large-scale flows considered in the present evaluations, it is difficult to determine the ground truth of the flows manually. Therefore, in evaluating the FPR performance of the models, flows from the following trustworthy sources were selected as normal instances and treated as the ground truth.

- DNS: Google DNS (8.8.8.8, 8.8.4.4), OpenDNS (208. 67.222.222, 208.67.220.220, 208.67.222.220, 208.67. 220.222), and Cloudflare/APNIC DNS (1.1.1.1, 1.0.0.1)

Table 6
Matching rates of characteristic rules.

Tag	Rate of IP addresses that match the rules
Botnet	35.1%
Exploit	7.35%
Tor	0.13%
Phishing	31.7%
Ransomware	0%
Malware	4.2%
Spam	6.55%
Cryptomining	0.07%
Scanner	47.73%

- Google search homepage: 142.251.35.174, 142.251.32.110, 142.250.80.78, 142.250.81.238, 172.217.13.206, 216.58.223. 110, 216.58.200.46, and 172.217.160.78

In the present study, the severity (S) is a numerical value in the range of 1 to 10. Hence, the performance of the random forest regressor was evaluated by computing the mean-squared error (MSE) of the regression results, defined as the mean of the square of the difference between the predicted and original severity values of each instance.

6.2. Assessment

6.2.1. Tag consistency with characteristic rules

The characteristic rules listed in Table 5 were used to examine whether the flows with the tag assigned in the CTI behaved as expected in the real-world sighting. If any of the flows from an IP address matched the rule(s) of any tags, the IP address was attributed to that tag(s). The matching rate for each tag was evaluated for every hour, and the mean matching rate for each tag was then computed. The corresponding results are presented in Table 6. It is seen that the actual activities of the flows are mostly not as expected from the tags.

To compensate for the possibility that the characteristics simply just happened not to appear within the observed one-hour period, the evaluation procedure was extended to 1 day and 12 days, respectively. The results presented in Fig. 6 indicate that, while the matching rate generally increased slightly for the different tags, most of the flows still did not match the corresponding characteristic rules. In other words, over 50% of the flows do not exhibit the behaviors expected from the characteristic rules.

Note that IP addresses are volatile identifiers in several cases. One common case is that a malicious source is behind an NAT device (e.g., a typical client host), and obtains its IP address through DHCP. While the source obtains a dynamic private IP address, it comes with a fixed public IP address (i.e., that of the NAT device) in the target environment. Thus, analysis of such hosts is similar to those with fixed public IP addresses. The presence of a malicious source in the case of tunneling or VPN also comes with a fixed public IP address in the target environment. Accordingly, analysis of such cases is also similar to that of NAT. However, if the malicious source obtains public IP addresses from DHCP, it will be difficult to tell such IP addresses from fixed IP addresses from perspective of the target environment in a scalable manner. This limitation is common to blacklisted IP addresses in the CTI and the present study. Accordingly, it may be one of the reasons that account for the low matching rates observed herein.

6.2.2. Clustering for each tag

To investigate the major activities of the flows with the same tag, the flows with each tag were clustered by DBSCAN in accordance with the 19 features described in Section 4.2. As shown in Fig. 7, each tag showed one major cluster which was obviously larger than any of the other clusters for the same tag. A manual inspection of the largest clusters showed that, for each tag, most of the flows contained only one packet or had an average packet payload of less 64 bytes (see Fig. 8).

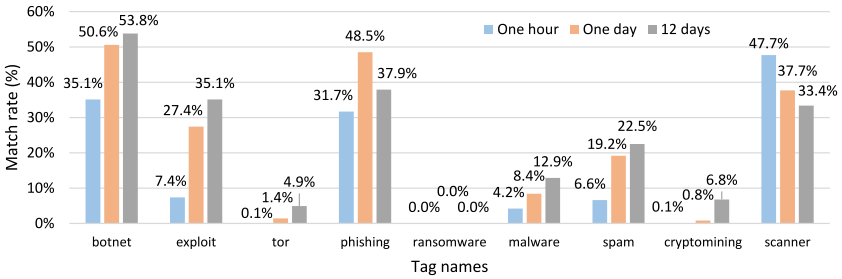


Fig. 6. Matching rates of characteristic rules.

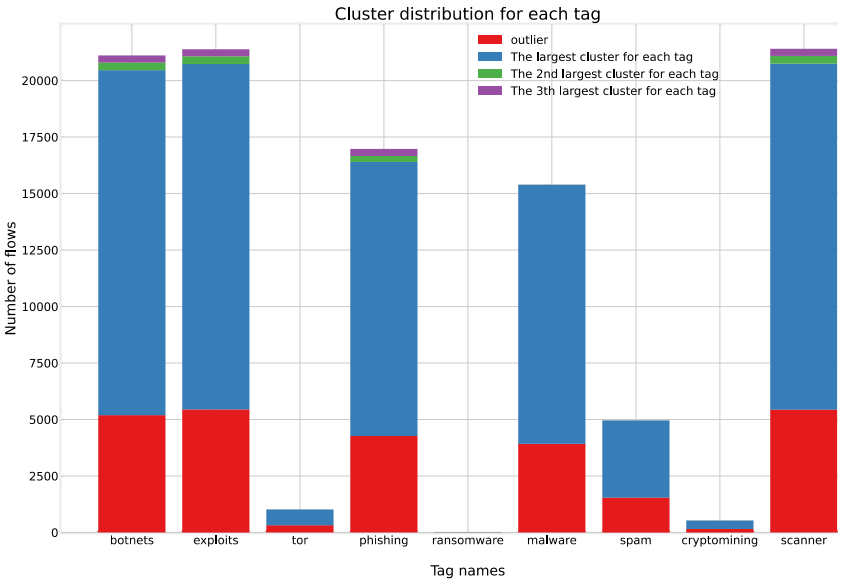


Fig. 7. Clustering results for each tag.

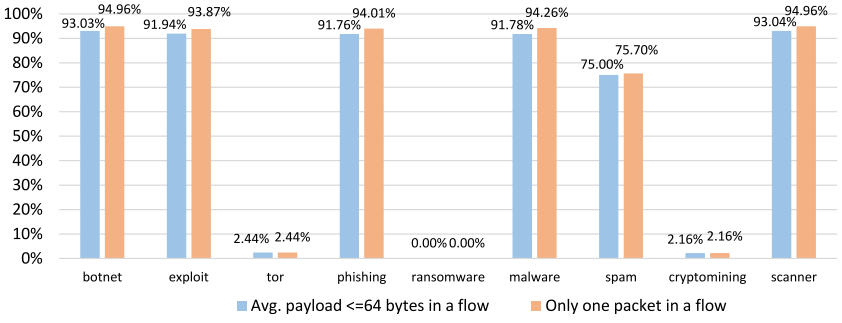


Fig. 8. Observations of the largest cluster for each tag.

The ransomware tag contained only a single flow, but most of the flows with the tor, spam and cryptomining tags have multiple packets. Nonetheless, over 90% of the overall flows had an average payload shorter than 64 bytes, a manual inspection of randomly sampled flows from each tag showed that the packets in the flows appeared to be probing or scanning packets.

6.2.3. Comparison with the IPS detection

The prevalence of probing or scanning packets among the NetFlow data suggests that many attacks may have been blocked by the IPS of the campus. To investigate this issue further, an analysis was performed of the one-day IPS records associated with the blacklisted IP addresses in the firewall and CTI platform. It is noted that the IPS was deployed between the campus network and the Internet, whereas only the flows from or to hosts in the considered department appear in the sighting.

Fig. 9 illustrates the relationship between the IP addresses detected by the IPS and CTI device, respectively, on the same day. The results presented on the left side of the figure show that out of all the IP addresses blocked by the IPS, flows from 339 of these addresses still appeared in the target environment. A total of 160 IP addresses appeared in the CTI blacklist on that day. In other words, even though some flows from malicious IP addresses were likely blocked from entering the campus by the IPS, a certain number of flows still managed to gain access to the target network. This suggests that the IPS detection results may have included false negatives, or some of the flows from the malicious IP addresses exhibited normal behavior. However, among these 339 IP addresses, only 10 (3%) fell within the blacklist of the intelligence-based firewall. Therefore, it was surmised that the blacklist on the CTI device may not have been updated in time to catch the latest malicious IP addresses. The results presented on the right side of

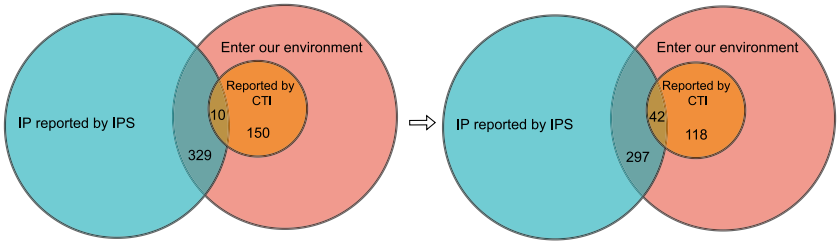


Fig. 9. Venn diagrams between IP addresses blocked by IPS and blacklist based on CTI.

Table 7
Severity distribution of IPS blocked sources in CTI.

Items	Low	Medium	High
Severity distribution of the IP addresses reported by the IPS in our sighting	156	132	51
Severity distribution of the IP addresses reported by the firewall	0	2	8
Severity distribution of the IP addresses after the blacklist has been updated	0	2	40

the figure show that the number of blacklisted IP addresses increased to 42 (12%) after two months. However, 120 IP addresses reported by the CTI were not reported by the IPS. This finding suggests that a large proportion of the attacks reported by the CTI were not blocked by the IPS. Thus, the low matching rate of the characteristic rules were not attributed to the blocking by the IPS, and the probing or scanning packets were likely what they were from the malicious sources, rather than the remaining packets that were not blocked by the IPS.

The IPS records, and their intersection with the CTI blacklist, were additionally examined from the perspective of severity. The corresponding results are presented in Table 7, where the first row presents the distribution of the severity levels of the 339 IP addresses that entered the target environment. (Note that the low level corresponds to 1 to 4, while the medium and high levels correspond to severities of 5 to 7 and 8 to 10, respectively). The second row of the table shows the distribution of the severity levels of the 10 IP addresses common to both the IPS blacklist and the CTI blacklist. Finally, the third row shows the distribution of the severity levels of the 42 IP addresses common to the two blacklists after the CTI blacklist was updated.

Under the assumption that the IPS detection results are correct, the firewall blacklist based on the CTI has a false negative rate of 97.1% (329/339). Even after the CTI blacklist is updated, the false negative rate is still 87.6% (297/339). Notably, the number of blacklisted IP addresses with high severity increases from 15.7% to 78.4% (from 8/51 to 40/51) after the CTI update. In other words, the update process improves the ability of the blacklist to detect most of the IP addresses with high severity. Although the blacklist continues to miss many IP addresses with low or medium severity, it is possible that some of these addresses are actually false positives in the IPS detection process, which accounts, at least partly, for the "failure" of the CTI device to capture them.

6.3. Augmentation

In general, the assessment results presented above show that the CTI-based blacklist results in a high false negative rate (FNR). Therefore, in the augmentation process, machine learning is leveraged to identify further flows similar to known malicious flows, and some of the corresponding flow sources are then added to the CTI blacklist. Notably, the augmentation process involves not only identifying more malicious IP addresses, but also revising their tags and severities.

Table 8
FNR and FPR of malicious flow detection model.

Metrics	Value
FNR	0.104
FPR of DNS flows	0.002
FPR of Google flows	0.002

Table 9
FNR of the tag prediction model for each tag.

Tag	FNR
Botnet	0.095
Exploit	0.097
Tor	0.645
Phishing	0.099
Ransomware	NaN
Malware	0.097
Spam	0.098
Cryptomining	0.722
Scanner	0.112

6.3.1. Learning models for augmentation

Malicious flow detection model The malicious IP list was obtained from the CTI device, and the individual flows were then labeled as malicious if they came from an IP address was on the list. Following the labeling process, 15-day malicious flows (around 2 million in total) were used to train the one-class SVM model (see Section 4.4). Finally, 8-day malicious flows (around 1.3 million) were used to estimate the FNR, and 1-day DNS flows (approximately 1.7 million) together with 15-day flows from the Google homepage (around 1.3 million) were used to estimate the FPR. Both the FPR and the FNR were calculated based on the total number of flows. The corresponding results are presented in Table 8.

Tag prediction models A one-class SVM model was trained for each tag. The IP list with tag information from the CTI was used to classify the source IP addresses of the flows into nine tagged datasets (one for each attack type). Some of the IP addresses had multiple tags, and hence some of the IP addresses were duplicated in several tagged datasets. After the prediction process, only the data with at least with one tag remained. Hence, the FNR of each tag model was calculated by dividing the total number of flows that the model predicted to contain a false tag by the total number of flows labeled with the tag.

As shown in Table 9, with the exception of the tor, ransomware, and cryptomining models, most of the models showed a low FNR. The poor performance of the three models is due, most probably, to the lack of flow data for training purposes (e.g., only one flow for the ransomware tag). Apart from the scanner tag model, all of the other models had a FNR of less than 0.1. In other words, the models were capable of identifying the tags of most of the malicious IP address flows.

Note that although generating more related network flows (e.g., by running malware samples in a sandbox) for training the tor, ransomware, and cryptomining models might address the issue of low FNRs in the models, the flows in the sighting were still relatively

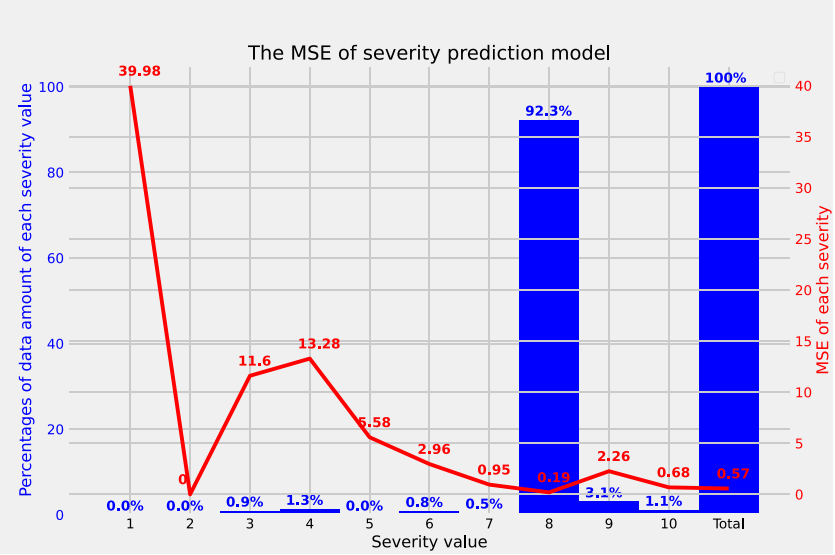


Fig. 10. The MSE and data distribution of the severity prediction model.

Table 10
Number and percentage of malicious IP addresses predicted for each severity level and tag.

Tag	Low		Medium		High	
	Number	Percentage	Number	Percentage	Number	Percentage
Botnets	2	1.85%	51	47.22%	73	67.59%
Exploits	2	1.85%	43	39.81%	54	50%
Phishing	4	3.7%	195	180.56%	122	112.96%
Malware	4	3.7%	94	87.04%	164	151.85%
Spam	2	1.85%	57	52.78%	77	71.30%
Scanner	1	0.93%	47	43.52%	50	46.30%
Total	4	3.7%	221	204.63%	169	156.48%

few for useful testing in practice. Moreover, since the present study focuses on correlation of the CTI and the sighting, the dataset in the evaluation is supposed to be collected in the real-world sighting, and thus using manually generated network traffic is beyond the discussion of this study, even though it may complement the present study for augmentation.

Severity prediction model Fig. 10 shows the MSE and data distribution of the severity prediction model results. As shown, the majority of the data flows had a predicted severity of 8 and were associated with a relatively low MSE. Although the MSE for the data flows with low severity were significantly larger, the prediction error was not regarded as a particular problem due to the low number of flows involved.

6.3.2 Prediction of malicious IP addresses

The network data collected over a period of six hours were used as a testing set to predict the malicious IP addresses using the three models described above. The CTI blacklist identified 108 malicious IP addresses over the corresponding observation period. Table 10 shows the number and percentage of malicious IP addresses predicted at each severity level for the six tags with higher detection performance (see Table 9). Note that since the six tags are predicted by six separate models, an IP address may be multi-tagged in the original CTI. According to the results, there were 3%, 205%, and 156% more predicted IP addresses in the low, medium, and high severity levels, respectively, than in the original CTI blacklist. Thus, the ability of the learning framework to identify more IP addresses with similar features to known malicious addresses is confirmed (particularly for IP addresses with medium or high severity).

To confirm practical feasibility of the proposed models, a publicly available reporting platform, AbuseIPDB (www.abuseipdb.com), was also checked to see whether any of the predicted IP addresses were also reported in AbuseIPDB in the 10 days prior to or after the prediction data. In particular, the AbuseIPDB records were used to estimate a lower bound for the accuracy of our predicted data. As shown in Table 11, 53% of the predicted IP addresses with high severity level were also reported by AbuseIPDB. Similarly, 25% and 21% for the low- and medium-severity IP addresses, respectively, were also reported in both platforms. In other words, the IP addresses predicted by the present models to be of high severity are indeed likely to be malicious sources and can thus be considered as candidates for augmenting the CTI blacklist.

Table 11
Rates of predicted IP addresses of different severity levels also reported in AbuseIPDB.

Type	Low	Medium	High
Botnets	50%	35%	48%
Exploits	50%	40%	54%
Phishing	25%	18%	55%
Malware	25%	37%	52%
Spam	50%	35%	48%
Scanner	0%	39%	56%
Total	25%	21%	53%

Table 12
Rate of augmented IP addresses of different severity levels.

Predicted severity level	The rate of augmented IPs to those from CTI	The rate that had been reported in AbuseIPDB
High	156% (169:108)	53% (89 of 169)
Medium	205% (221:108)	21% (45 of 221)
Low	3.7% (4:108)	25% (1 of 4)

Table 12 collates the results presented in Tables 10 and 11. The results confirm that the data with a higher severity are those that require the most attention. Moreover, the data with a lower severity level, as mentioned above, are likely to be overestimated.

6.4 Summaries of main observations

The main observations from the assessment experiments can be summarized as follows:

- For the false positives cases, more than 50% of the flows did not match the characteristic rules for the associated tag. Furthermore, the mismatch percentage exceeded 90% for some of the tags. In

other words, the tags listed in the CTI did not precisely match the actualities observed in the sighting.
- Even though many of the malicious sources did not present the behavior expected of the corresponding tags, they still exhibited obvious probing or scanning behaviors.
- A significant inconsistency was observed between the IP blacklist obtained from the CTI and the malicious IP addresses reported by the IPS (a false negative rate of 97.1%). However, when the CTI blacklist was updated using some of the malicious sources not originally in the list, the false negative rate fell to 87.6%, with the majority of this reduction associated with high-severity sources.

The main observations for the augmentation experiments are as follows:

- Twenty percent of the predicted tags did not appear in the original CTI blacklist, but were successfully detected by the proposed learning models.
- Eighty-nine of the predicted high-severity IP addresses were also reported by AbuseIPDB, where these addresses accounted for 53% of the total number of predicted high-severity IP addresses. In other words, the learning framework proposed in this study successfully captures malicious IP addresses blacklisted elsewhere. The predicted high-severity IP addresses were 156% more than those reported by the CTI blacklist over the observation period.

7 Conclusion and future work

Given the increasing importance of CTI in helping identify malicious events in network environments, it is essential to assess the consistency of the CTI with the actual threats observed in the sightings. Accordingly, the present study has proposed a systematic approach for not only assessing the CTI in terms of the false positives and false negatives in the sighting, but also augmenting the CTI considering the false negative outcomes. The present study found that over 50% of the flows do not match the characteristic rules for the associated tag, and many of the malicious sources exhibited obvious probing or scanning behaviors. The IP blacklist obtained from the CTI was also found inconsistent with the malicious IP addresses reported by the IPS (a false negative rate of 97.1%). For augmentation, the proposed learning framework can capture malicious IP addresses blacklisted elsewhere. The high-severity IP addresses identified in the sighting were 156% more than those originally blacklisted by the CTI over the observation period.

Future studies will conduct deeper investigations into the proposed CTI assessment and augmentation mechanisms. For example, further assessment trials will be performed to observe the sightings in multiple network environments to be more comprehensive. Meanwhile, further augmentation experiments will also be conducted to evaluate the precision and generalizability of the learning models, with particular emphasis on the preprocessing of the data (including feature extraction and selection). In addition, more complex models will be considered for prediction purposes, such as time series models.

CRediT authorship contribution statement

Po-Ching Lin: Conceptualization, Methodology, Writing – review & editing. **Wen-Hao Hsu:** Software, Investigation, Writing – original draft. **Ying-Dar Lin:** Conceptualization, Validation, Supervision. **Ren-Hung Hwang:** Conceptualization, Validation, Supervision. **Hsiao-Kuang Wu:** Validation, Supervision. **Yuan-Cheng Lai:** Validation, Supervision. **Chung-Kuan Chen:** Validation, Resources.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data that has been used is confidential.

Acknowledgments

This work was supported in part by National Science and Technology Council (NSTC), Taiwan, and Cycraft Technology.

References

[1] World Economic Forum, The global risks report, 2021, URL https://www3.weforum.org/docs/WEF_The_Global_Risks_Report_2021.pdf. (Published 19 January 2021).
[2] P. Gao, F. Shao, X. Liu, X. Xiao, Z. Qin, F. Xu, P. Mittal, S.R. Kulkarni, D. Song, Enabling efficient cyber threat hunting with cyber threat intelligence, in: IEEE 37th International Conference on Data Engineering, ICDE, 2021, pp. 193–204, http://dx.doi.org/10.1109/ICDE51399.2021.00024.
[3] G. Farnham, Tools and Standards for Cyber Threat Intelligence Projects, White paper from SANS Institute, 2013.
[4] National defense authorization act for fiscal year 2006, 2006, URL https://www.govinfo.gov/content/pkg/PLAW-109publ163/html/PLAW-109publ163.htm. (Published 6 January 2006).
[5] R. Hofstede, P. čeleda, B. Trammell, I. Drago, R. Sadre, A. Sperotto, A. Pras, Flow monitoring explained: From packet capture to data analysis with NetFlow and IPFIX, IEEE Commun. Surv. Tutor. 16 (4) (2014) 2037–2064, http://dx.doi.org/10.1109/COMST.2014.2321898.
[6] E.B. Claise, Cisco Systems NetFlow Services Export Version 9, RFC 3954, 2004, RFC Editor, URL https://www.rfc-editor.org/rfc/rfc3954.txt.
[7] B.T. B. Claise, P. Aitken, Specification of the IP flow information export (IPFIX) protocol for the exchange of flow information, 2013, URL https://www.ietf.org/rfc/rfc7011.txt, RFC 7011.
[8] B.-R. Chen, Z. Liu, J. Song, F. Zeng, Z. Zhu, S.P.K. Bachu, Y.-C. Hu, FlowTele: Remotely shaping traffic on internet-scale networks, in: Proceedings of the 18th International Conference on Emerging Networking EXperiments and Technologies, CoNEXT, 2022, pp. 349–368, http://dx.doi.org/10.1145/3555050.3569139.
[9] R. Mills, A.K. Marnerides, M. Broadbent, N. Race, Practical intrusion detection of emerging threats, IEEE Trans. Netw. Serv. Manag. 19 (1) (2022) 582–600, http://dx.doi.org/10.1109/TNSM.2021.3091517.
[10] S. Lee, H. Cho, N. Kim, B. Kim, J. Park, Managing cyber threat intelligence in a graph database: Methods of analyzing intrusion sets, threat actors, and campaigns, in: International Conference on Platform Technology and Service, PlatCon, 2018, pp. 1–6, http://dx.doi.org/10.1109/PlatCon.2018.8472752.
[11] Y. Gao, X. Li, H. Peng, B. Fang, P.S. Yu, HinCTI: A cyber threat intelligence modeling and identification system based on heterogeneous information network, IEEE Trans. Knowl. Data Eng. 34 (2) (2022) 708–722, http://dx.doi.org/10.1109/TKDE.2020.2987019.
[12] Y. Gao, X. Li, J. Li, Y. Gao, N. Guo, Graph mining-based trust evaluation mechanism with multidimensional features for large-scale heterogeneous threat intelligence, in: IEEE International Conference on Big Data, Big Data, 2018, pp. 1272–1277, http://dx.doi.org/10.1109/BigData.2018.8622111.
[13] K.B. Mavzer, E. Konieczna, H. Alves, C. Yucel, I. Chalkias, D. Mallis, D. Cetinkaya, L.A.G. Sanchez, Trust and quality computation for cyber threat intelligence sharing platforms, in: IEEE International Conference on Cyber Security and Resilience, CSR, 2021, pp. 360–365, http://dx.doi.org/10.1109/CSR51186.2021.9527975.
[14] Y.-T. Huang, C.Y. Lin, Y.-R. Guo, K.-C. Lo, Y.S. Sun, M.C. Chen, Open source intelligence for malicious behavior discovery and interpretation, IEEE Trans. Dependable Secure Comput. 19 (2) (2022) 776–789, http://dx.doi.org/10.1109/TDSC.2021.3119008.
[15] R. Azevedo, I. Medeiros, A. Bessani, PURE: Generating quality threat intelligence by clustering and correlating OSINT, in: 18th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/13th IEEE International Conference on Big Data Science and Engineering, TrustCom/BigDataSE, 2019, pp. 483–490, http://dx.doi.org/10.1109/TrustCom/BigDataSE.2019.00071.
[16] M. Sills, P. Ranade, S. Mittal, Cybersecurity threat intelligence augmentation and embedding improvement - a healthcare usecase, in: IEEE International Conference on Intelligence and Security Informatics, ISI, 2020, pp. 1–6, http://dx.doi.org/10.1109/ISI49825.2020.9280482.
[17] V. Mavroeidis, R. Hohimer, T. Casey, A. Jesang, Threat actor type inference and characterization within cyber threat intelligence, in: 13th International Conference on Cyber Conflict, CyCon, 2021, pp. 327–352, http://dx.doi.org/10.23919/CyCon51939.2021.9468305.
[18] H. Suryotrisongko, Y. Musashi, A. Tsuneda, K. Sugitani, Robust botnet DGA detection: Blending XAI and OSINT for cyber threat intelligence sharing, IEEE Access 10 (2022) 34613–34624, http://dx.doi.org/10.1109/ACCESS.2022.3162588.

[19] A. Berady, M. Jaume, V.V.T. Tong, G. Guette, From TTP to IoC: Advanced persistent graphs for threat hunting, IEEE Trans. Netw. Serv. Manag. 18 (2) (2021) 1321–1333, http://dx.doi.org/10.1109/TNSM.2021.3056999.
[20] P. Krishnamurthy, F. Khorrami, S. Schmidt, K. Wright, Machine learning for NetFlow anomaly detection with human-readable annotations, IEEE Trans. Netw. Serv. Manag. 18 (2) (2021) 1885–1898, http://dx.doi.org/10.1109/TNSM.2021.3075656.
[21] SpeedGuide, SpeedGuide.net – the Broadband, 2022, URL https://www.speedguide.net.
[22] phaag, phaag/nfdump, 2022, URL https://github.com/phaag/nfdump.
[23] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, J. Mach. Learn. Res. 12 (2011) 2825–2830.

Po-Ching Lin (Member, IEEE) received the Ph.D. degree in computer science from the National Chiao Tung University, Hsinchu, Taiwan, in 2008. He joined the Faculty of the Department of Computer Science and Information Engineering, National Chung Cheng University, in August 2009. He is currently a Professor. His research interests include network security, network traffic analysis, and performance evaluation of network systems.

Wen-Hao Hsu received Master degree of Computer Science and Information Engineering from National Chung Cheng University, Taiwan in 2022. His research interests include network security and intrusion detection.

Ying-Dar Lin (Fellow, IEEE) is a Chair Professor of computer science at National Yang Ming Chiao Tung University (NYCU), Taiwan. He received his Ph.D. in computer science from the University of California at Los Angeles (UCLA) in 1993. He was a visiting scholar at Cisco Systems in San Jose during 2007–2008, CEO at Telecom Technology Center, Taiwan, during 2010–2011, and Vice President of National Applied Research Labs (NARLabs), Taiwan, during 2017–2018. He was the founder and director of Network Benchmarking Lab (NBL) in 2002–2018, which reviewed network products with real traffic and automated tools, and has been an approved test lab of the Open Networking Foundation (ONF). He also cofounded L7 Networks Inc. in 2002, later acquired by D-Link Corp, and O'Prueba Inc. a spin-off from NBL, in 2018. His research interests include network security, wireless communications, network softwarization, and machine learning for communications. His work on multi-hop cellular was the first along this line, and has been cited over 1000 times and standardized into IEEE 802.11s, IEEE 802.15.5, IEEE 802.16j, and 3GPP LTE-Advanced. He is an IEEE Fellow (class of 2013), IEEE Distinguished Lecturer (2014–2017), ONF Research Associate (2014–2018), and received in 2017 Research Excellence Award and K. T. Li Breakthrough Award. He has served or is serving on the editorial boards of several IEEE journals and magazines, including Editor-in-Chief of IEEE Communications Surveys and Tutorials (COMST, 1/2017–12/2020). He published a textbook, Computer Networks: An Open Source Approach, with Ren-Hung Hwang and Fred Baker (McGraw-Hill, 2011).

Ren-Hung Hwang (Senior Member, IEEE) received his Ph.D. degree in computer science from the University of Massachusetts, Amherst, Massachusetts, USA, in 1993. He is the Dean of the College of Artificial Intelligence, National Yang Ming Chiao Tung University (NYCU), Taiwan. Before joining NYCU, he was with National Chung Cheng University, Taiwan, from 1993 to 2022. He is currently on the editorial boards of IEEE Communications Surveys and Tutorials and IEICE Transactions on Communications. He received the Best Paper Award from The 6th International Conference on Internet of Vehicles 2019, IEEE Ubi-Media 2018, IEEE SC2 2017, IEEE IUCC 2014, and the IEEE Outstanding Paper Award from IEEE IC/ATC/ICA3PP 2012. He served as the general chair of the International Computer Symposium (ICS), 2016, and International Symposium on Pervasive Systems, Algorithms, and Networks (ISPAN) 2018, International Symposium on Computer, Consumer and Control (IS3C) 2018, IEEE DataCom 2019 (The 5th IEEE International Conference on Big Data Intelligence and Computing). His current research interests include in Deep Learning, Wireless Communications, Network Security, AIoT, and Cloud/Edge/Fog Computing.

Eric Hsiao-Kuang Wu (Member, IEEE) received the B.S. degree in computer science and information engineering from National Taiwan University, in 1989, and the master's and Ph.D. degrees in computer science from the University of California, Los Angeles (UCLA), in 1993 and 1997, respectively. He is currently a Professor of computer science and information engineering with National Central University, Taiwan. His research interests include wireless networks, mobile computing, and broadband networks. He is also a member of the Institute of Information and Computing Machinery (IICM).

Yuan-Cheng Lai received his Ph.D. degree in the Department of Computer and Information Science from National Chiao Tung University in 1997. He joined the faculty of the Department of Information Management at National Taiwan University of Science and Technology in August 2001 and has been a distinguished professor since June 2012. His research interests include performance analysis, software-defined networking, wireless networks, and IoT security.

Chung-Kuan Chen is currently a research director in CyCraft, who is responsible for organizing the research team, and Adjunct Assistant Professor in Soochow University, Taiwan. He earned his PHD degree of Computer Science and Engineering from National Chiao-Tung University (NCTU). His research focuses on cyber attack and defense, machine learning, software vulnerability, malware and program analysis. He also dedicates to security education. As the founder of NCTU hacker research clubs, he trained students to participate in world-class security contests, and has experience of participating DEFCON CTF (2016 in HITCON Team and 2018 as coach in BFS team). Besides, he has presented technical presentations in technique conferences, such as BlackHat, HITCON, HITB, RootCon, CodeBlue, FIRST and VXCON. As an active member in Taiwan security community, he is the chairman of HITCON review committee as well as a director of Association of Hacker In Taiwan, and member of CHROOT - the top private hacker group in Taiwan.